11th Canadian-International Conference on Composites

CONVERSION OF A PLC-BASED GRIPPER SYSTEM INTO A MODULAR, CROSS-LINKED PRODUCTION UNIT WITH INNOVATIVE CONTROL SYSTEM

Körber, M.[1]*, Krebs, F.[1]
[1] Center for Lightweight Production Technology (ZLP), German Aerospace Center (DLR), Augsburg, Germany
* Corresponding author (marian.koerber@dlr.de)

ABSTRACT

At the German Aerospace Center (DLR) in Augsburg (Germany) automation solutions for manufacturing processes of full scale aircraft structures are developed. Therefor a complex gripper system for the automation of a dry fibre preform process was developed based on a standardized PLC control system. Taking the concept of industry 4.0 into account, this system has been transformed into a cross-linked and flexible production unit, which can be operated using an innovative control systems. The network based process control is built on the exchange of process-relevant data using the OPC-UA interface and a Python-based protocol called PyADS. With the help of OPC UA this gripper system will be used as a platform for optimization algorithm and as an example for Industry 4.0 and Big Data applications.

KEYWORDS: OPC UA, Industry 4.0, data exchange, interoperability, process optimization

1 INTRODUCTION

production technology is changing towards a new era of digitized manufacturing. New technologies, such as modern communication systems, the Internet and processes based on human-machine cooperation, are leading to the fourth industrial revolution. This revolution was given the name Industry 4.0. It combines several novel approaches that are designed to make the production environment more agile and flexible.

The idea of industry 4.0 is based on the concepts like the Internet of Things (IoT) and Cyber-Physical Systems (CPS). With the help of these technologies, manufacturing processes can be created in which production participants are networked with each other and can thus exchange data with one another at any time. In contrast current automation systems are laid out hierarchical using the classic automation pyramid approach. The internet has made it possible to realize this data interchange across borders with the help of standardized interfaces. This exchange can be carried out from person to machine, from production machine to warehouse stock or from company to customer. [1][2]

This novel approach makes it possible to operate highly automated production systems, which not only provides available data but controls themselves independently based on available information. The challenges arising from this concept are to interconnect the connected machines with a minimum of development and integration effort. This leads to the necessity of standards in terms of data exchange protocols as well as syntax and semantics. Therefore, the OPC-Foundation developed a standard for data exchange as a platform-independent, service-oriented architecture called OPC Unified Architecture (OPC UA). [3]

The DLR in Augsburg conducts research for innovative manufacturing processes for fiber composite components. A wide variety of prototype systems are being developed and tested for this purpose. The application range extends through the entire production chain, from cutting the textile reinforcement layers to preforming the dry textiles and the infiltration of the component. All these process steps are controlled with the aid of data, which on the one hand are transferred from a central process control system to the automation cells. On the other hand, the systems generate a large amount of data during the process, which is recorded with the use of sensors. This data serves as parameters for the control system, but they can also be stored in a database for later reference and, if necessary, subsequently used as the basis for optimization and documentation. Different control approaches are used for the prototypes, depending on the requirements and area of application. Since the idea behind Industry 4.0 is to control distributed plants and store the collected data in a central data warehouse, an inhomogeneous control environment leads to a high level of integration effort and an equally high level of vulnerability to faults. For this reason, the control and communication structure used at DLR Augsburg is based on the OPC UA communication platform mentioned above.

In the first part of this paper, the concept and application of the OPC UA communication platform will be examined in more detail. The second part describes how to use the OPC UA interface to control the end effector called "Modular Gripper" and collect its process data.

2 OPC UA OVERVIEW

The standard OPC Unified Architecture (OPC UA) was developed to fulfill two tasks. On the one side it is supposed to allow distributed systems to exchange information using a client/server model. On the other side OPC UA also provides the possibility of information modeling which describes the system and the available information.

OPC UA offers important requirements for the implementation of an industrial 4.0-based control concept. This includes reliable and robust communication between distributed systems that can be achieved through robustness and fault tolerance. In order to integrate OPC UA interfaces in different conventional systems and at the same time guarantee high scalability, the OPC UA standard was developed platform independent based on a high variety of programming languages.

As mentioned above OPC UA is not only used to exchange data in an interoperable way, but also information with a defined semantic. This is achieved by assigning every piece of data to an address space model. This model consists out of nodes and dependencies between those nodes. To give the address space a specific structure different node classes were created, which are described with the help of attributes. A *NodeClass* is similar to an object from object-oriented programming. Following *NodeClasses* are available:

- *Objects* help to give a structure to the address space
- *Object types* describe the type of an object. It models hierarchy between OPC UA Nodes.
- *Variables* provide data that is defined through two attributes, the value and its data type.
- *Variable types* define the semantic of variables in order to structure the address space in case of multiple sub-variables.
- *Methods* define callable functions that can be called via the OPC UA interface
- *Data types* provide type definitions variable data
- *Reference type* describes the type of a reference.
- *Views* assure a simple way for clients to access node information.

Each node class has two attributes with which it can be identified inside the OPC UA address space. It is the *NodeID* and the *DisplayName*. The last attribute is commonly used to give the user a visual overview of the address space structure. [4]

Furthermore, events are supported that are raised in certain cases to react on Alarms & Events (AE) or Data Change events (DA). Among other things, an event has a reception time, a message and a severity level.

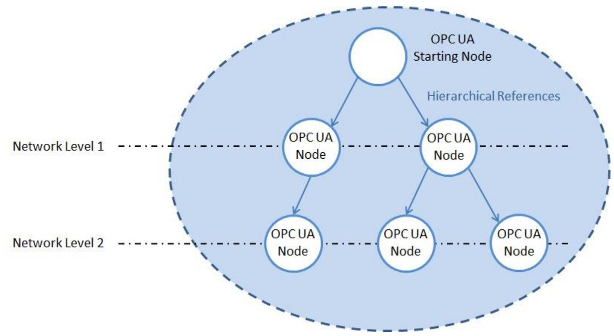

Figure 1: Representation of the hierarchically structured address space

3 ADAPTION OF AN PLC-BASED SYSTEM TO AN OPC UA BASED CONTROL SYSTEM

The Modular Gripper is an endeffector system for the automated draping of dry textile carbon fiber cut pieces (Figure 2). Therefore, it can actively change its gripping geometry to adapt to the moulds geometry. It follows a vortex-rib structure. The vertebra runs centrally from the upper to the lower end of the gripper. The elements are supported by two glass fiber rods. Three linear actuators are applied to the different positions of the glass fiber rods. Through the specific control of the actuators, a shape can be applied to the rods, which is in turn transferred to the gripper. The 15 rib units provide the second curvature to the system. The modular gripper has 127 electrically operated vacuum units for handling textile cut pieces. Those units have can generate a very high-volume flow with which air-permeable fabrics can be reliably gripped. Each of the vacuum units can be operated individually. Their gripping force can be adjusted in eight intensities. This gives the operator the possibility to adjust the holding force on the fiber material and thus directly influencing the draping result. These results are monitored by a sensor system with 20 sensors integrated in the gripper surfaces. This system detects relative movement between the textiles and the suction surfaces. In total the Modular Gripper is operated with 145 parameters which make this system to one of the most complex ones at DLR Augsburg. [5][6]

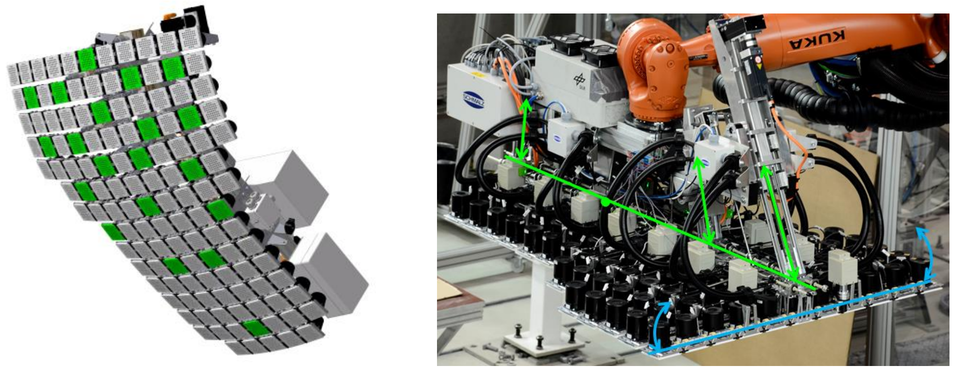

Figure 2: Left: Deformed gripper system with 20 sensor equipped suction surfaces; Right: Functional schematic of the spine (green) and rip (blue) system

3.1 Original PLC-based control setup

The original control setup is based on a TwinCAT2 PLC system. On the hardware side the system consists of a Beckhoff Embedded-PC, on which the operating system and the control software are running. In addition to the Embedded-PC, several additional components, such as fieldbus modules or motor controllers, can be attached. In the original setup, the system was developed to execute the entire process with as little external input as possible. Therefore, the entire process is divided into individual steps, which are executed chronologically. The only way to take online influence of the process is via an EtherCAT-bus system that directly connects the grippers TwinCAT Embedded-PC and the KUKA KRC unit, which is the control unit of the robot system. The data exchange between both units is limited to a pure boolean status notification. That means the KRC starts a process step and the grippers control system is reporting a successful execution or a possible error event. The process steps and their chronological order are as follows:

1. Set gripper to initial configuration
2. Activate suction units to grip cut piece in flat state
3. Deform and drape the textile according to predefined draping configuration
4. Deactivate suction units to grip cut piece in flat state

A direct influence on the process sequence or a repetition of a process step is not possible due to the rigid system. Another disadvantage of the original control system is the adaptation of the process parameters. This includes the target motor positions of the three spine actuators as well as those of the 15 rib actuators. In addition, there are the parameters of 127 gripping force intensities. These parameters must be manually specified offline for each cut-piece using a TwinCAT-specific graphical user interface. An online adjustment of these parameters during the process is not possible. This procedure is very problematic in a research environment, since this gripper system is intended to serve as a platform for innovative, dynamic optimization processes.

The data exchange between the sensor system and the evaluation software must be carried out in parallel with direct TCP communication. The exchanged sensor data consists out of a stream of strings that carry the information in a specific format. The sensor analysis software that is running on a standard PC interprets the string data and assigns it to one of the 20 sensors. This leads to an even higher system complexity as two different communication systems must be operated in parallel.

3.2 OPC UA-based control setup

The goal of the conversion from a mainly PLC controlled system to an OPC UA based control system is to make all functions accessible online using a set of OPC UA variables. On the one hand, this should considerably simplify process control, but on the other hand it should also provide the user with better and more flexible system accessibility. In this way, it should be ensured that the process can be controlled and optimized online and that all collected data such as current motor positions, current gripping force intensities, activity status and sensor data can be published online to the OPC UA network.

The hardware of the control system should not be changed if possible, since the Beckhoff components installed are designed to control many motors. The adaptations will mainly relate to the software. In order to be able to continue using Beckhoff components, a new, leaner PLC software must be developed. Its task is to provide an interface between the input of the motor and suction controllers and the variables to be published to the OPC UA network. That means the input variable values for motor and suction units must be interpreted and converted into a predefined format that Beckhoff controllers are able to work with.

The hardware platform of the new OPC UA control setup is a Raspberry Pi that is used as the OPC UA server. There are three individual tasks running on this Raspberry Pi (see Figure 3):

- The *OPC UA server* defines the OPC UA address space structure and publishes all changes in the OPC UA network.
- *PyADS* is a Python wrapper for TwinCATs ADS library. It creates an interface between the PLC controller and the Raspberry Pi as the build-in Embedded-PC is not able to run an own OPC UA Client.
- *I^2C bus interface* to gather the sensor data.

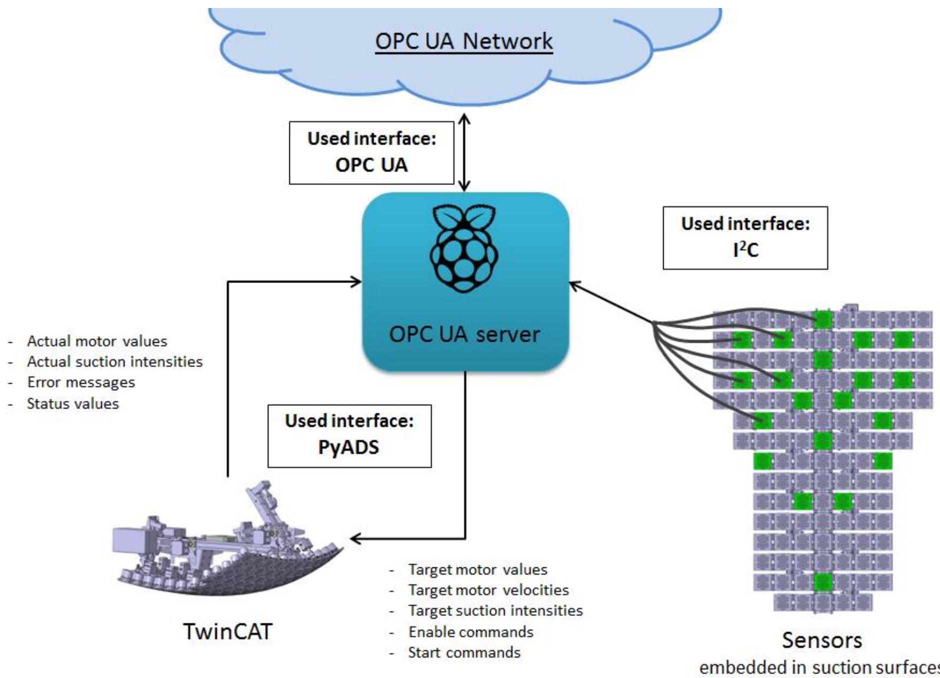

Figure 3: Data exchange setup based on OPC UA

The first step to configure OPC UA server, is to define the OPC UA address space model. For this purpose, objects with specified hierarchy levels must be created. The objects describe individual functions of the system. In the case of the Modular Gripper, four main functions are defined:

- Spine deformation
- Rip deformation
- Suction surface sensor system
- General system functions

As described in figure Figure 1 the address space model is initialized with a starting node. Each function will be represented by an *ObjectNode* that is set to the next hierarchy levels and attached to the starting node. The next step is to set up *NodeVariables* that will transport the intended information. Figure Figure 4 gives an overview of the Modular Grippers address space structure.

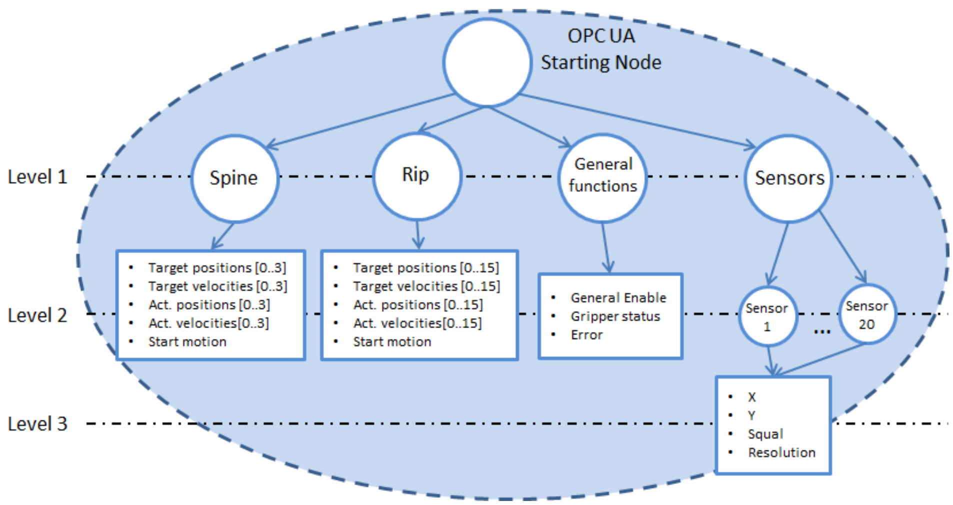

Figure 4: The Modular Grippers OPC UA address space

The variables are selected in such a way that each function of the gripping system can be addressed and all generated information can be accessed. The variables are selected in such a way that each function of the gripping system can be addressed and all generated information can be accessed. The spine and rib groups each have variables for the nominal motor positions and the nominal speed as well as the current motor positions and speeds. The data type used is an array with a certain number of integer or double placeholders corresponding to the number of motors. For the spine function three and for the ribs 15 placeholders. After setting these values the boolean variable “start motion” triggers the deformation of the suction surface. The structure of the sensor model goes one level deeper. Below the basic sensor node, an *ObjectNode* is created for each sensor. These objects each contain the variables for all data contained by the sensors (X, Y, Squal-Value, Resolution). For this development we decided to use the event-based data exchange option. That means that an event will be triggered as soon as a variable was changed by a client. The server algorithm decides what effect this will have. One example would be that the motors start variable is set TRUE by any connected client. This information will be transported with the help of the PyADS module to the grippers PLC and causes it to move the motors to the target motor position.

Due to the platform independence of the OPC UA interface, control commands and data acquisition can be performed with any device capable of executing an OPC UA client. This can be the KRC robot controller, which sets the variables in the process program by setting OPC UA and thus obtains direct control over the functions of the gripping system.

4 CONCLUSION

The OPC UA based control and data exchange interface has several significant advantages over the original and PLC-only control. The OPC UA interface allows a much more flexible and intuitive control. This enables the user to adjust every gripper function online without having to pay attention to fixed process sequences. The process logic can be operated from a non-PLC-based system outside of a TwinCAT environment. This enables the use of external process optimization software in the future.

This last point is the main advantage of the OPC UA interface. Because of the OPC UAs interoperability every device that is able to run an OPC UA client and that is connected to the network, is able to get access to the control variables and its information. This enables applications that, for example, store all collected data in one or more databases and thus provide a database for optimization and documentation. At DLR, this function was carried out using the MongoDB and InfluxDB databases.

To sum up, the gripper system is now ready to be operated in an flexible, research environment.

5 FUTURE WORK

The future work will consist in developing further OPC UA client applications, which enable a more comfortable and GUI-based control. On the one hand, this can be implemented with the help of specially adapted PC software; on the other hand, a web-based application is also suitable for such an application. This would be executable with the help of a browser from any web-enabled device without the need to install special software.

For the process of automated draping different optimization applications as external software already exist. These programs could be equipped with an OPC UA client. With the help of the OPC UA network, these tools have a large database as they have access to all available data and can also actively intervene in the process.

6 ACKNOWLEDGEMENTS

We would like to thank all colleagues involved for their support in this project.

REFERENCES

[1] S. Cavalieri, M. S. Salafia, and S. Scroppo, “Integrating opc ua with web technologies to enhance interoperability,” vol. 61, 01 2019.

[2] U. Automation, OPC UA steuert das Fenster zum Universum, 2012 (accessed May 23, 2019). [Online]. Available: https://www.unified-automation.com/ de/referenzen/case-tudies/nachricht-detailansicht/article/ opc-ua-controls-the-window-to-the-universe.html

[3] W. Mahnke, S. H. Leitner, and M. Damm, OPC Unified Architecture. Springer Verlag, 2009.
[4] OPCFoundation, OPC UA Part 3: address space model specification, release 1.04, OPC Foundation, 2017.
[5] H. Voggenreiter and D. Nieberl, eds. AZIMUT Abschlussbericht. Augsburg: TIB, 2015.
[6] M. Körber and C. Frommel, eds. Sensor-Supported Gripper Surfaces for Optical Monitoring of Draping Processes. Stuttgart: SAMPE, 2017